INFORMATION EXTRACTION FROM BROADCAST NEWS SPEECH DATA

David D. Palmer[1,2] *palmer@mitre.org*

John D. Burger[1] *john@mitre.org*

Mari Ostendorf[2] *mo@raven.bu.edu*

[1]*The MITRE Corporation* [2]*Boston University*

ABSTRACT

In this paper we describe a robust algorithm for information extraction from spoken language data. Our probabilistic algorithm builds on results in language modeling, using class-based smoothing to produce state-of-the-art performance for a wide range of speech error rates. We show that our system performs well with sparse data, as well as with out-of-domain data.

1. INTRODUCTION

Extracting linguistic structure such as proper names, noun phrases, and verb phrases is an important first step in many systems aimed at automatic language understanding. While significant progress has been made on this problem, most of the work has focused on "clean" textual data such as newswire texts, where cues such as capitalization and punctuation are important for obtaining high accuracy results.

However, there are many data sources where these cues are not reliable, such as in spoken language data or single-case text. Spoken language sources, in particular, pose additional problems because of disfluencies and speech recognition errors. This paper addresses the problem of information extraction from speech, introducing a new probabilistic approach that builds on language modeling techniques to obtain robust results even for high error rate tasks. Previous approaches that have addressed speech data have consisted largely of applying an existing text-based system to speech data, ignoring the fact that information is both lost (due to recognition errors) and gained (from acoustic cues and word confidence prediction) when moving from text to speech.

We have developed a probabilistic framework for the identification of linguistic structure in spoken language data. Our model builds on the work of BBN's Identifinder system (Bikel et al., 1997, Bikel et al., 1999), which uses a hidden state sequence to represent phrase structure and state-conditioned word bigram probabilities as in a hidden Markov model. The BBN model incorporates non-overlapping features about the words, such as punctuation and capitalization, in a bigram back-off to handle infrequent or unobserved words. Viterbi-style decoding is used to produce the most likely sequence of phrase labels in a test corpus. The simple Identifinder approach has resulted in high performance on many text-based tasks, including English and Spanish newswire texts.

In this work we describe our approach to the problem of information extraction for spoken language data. A key component of our approach is that infrequent data is handled using a class-based smoothing technique (Iyer & Ostendorf, 1997) that, unlike the feature-dependent back-off, allows for ambiguity of word classes. Thus, we can incorporate information from place and name word lists, as well as simple part-of-speech labels, and account for the fact that some words can be used in multiple classes.

The specific information extraction task we address in this work is name recognition (identifying names of persons, locations, and organizations), as well as identification of temporal and numeric expressions. Also known as *named entities* (NEs), these phrases can be useful to identify in many language understanding tasks, such as coreference resolution, sentence chunking and parsing, and summarization/gisting. Named entity identification in written documents has been examined extensively under the auspices of the Message Understanding Conferences (MUC) sponsored by DARPA, and performance of name recognizers, or named entity taggers, on written documents such as *Wall Street Journal* articles is comparable to human performance (usually 94–96% accuracy[1]). DARPA has recently expanded the scope of its information extraction evaluations to include named entity recognition in speech data, both conversational speech and broadcast news speech.

We show our approach to produce high performance on speech data with a wide range of word error rates (WERs), including reference transcriptions (WER 0%) of broadcast news. In the official 1998 DARPA Hub-4 information extraction evaluation, our phrase model produced excellent results when

[1] Named entity system performance is typically reported in terms of the *F-measure*, which is the harmonic mean of recall and precision. All system accuracies we report will be in terms of the F-measure.

applied to the task of identifying names in broadcast news transcripts. When applied to transcripts generated by automatic speech recognition, the model showed high performance (71–81% accuracy), despite word error rates ranging from 13% to 28%. Our model also produced excellent results (88% accuracy) in recognizing named entities in the reference transcripts.

2. SYSTEM OVERVIEW

The mathematical model we use is very similar to the model developed by BBN (Bikel et al., 1997), with several important differences. In Sections 2.1–2.3 we will review the basic assumptions made in the model and discuss the differences between our approach and BBN's work.

2.1. Probabilistic Model

As in the BBN model, we use a hidden state sequence to represent phrase structure, with the assumption that each word in a document is emitted by a state in the model, similar to a hidden Markov model (HMM). The problem is thus framed as a maximization of the hidden state sequence ($s_1 \ldots s_L$) most likely to have produced the known word sequence ($w_1 \ldots w_L$), or $P(s_1 \ldots s_L | w_1 \ldots w_L)$.

This probability can be reformulated using Bayes' Law and the chain rule; making a Markov assumption that the state at time t is dependent only on the state and observation at time $t-1$, we arrive at the following formulation:

$$\begin{aligned} &\arg\max_{s_1 \ldots s_L} P(s_1 \ldots s_L \mid w_1 \ldots w_L) \\ &\approx \arg\max_{s_1 \ldots s_L} \left(\prod_{t=1}^{L} P(w_t \mid s_t, w_{t-1}) P(s_t \mid s_{t-1}, w_{t-1}) \right) \end{aligned}$$

The two probabilities in the maximization can then be optimized separately. The first, $P(w_t|s_t,w_{t-1})$ can be thought of as a state-dependent bigram language model; the second, $P(s_t|s_{t-1},w_{t-1})$, can be viewed as the state transition of an HMM, from the state at time $t-1$ to the state at time t. It is important to note that, while we make the Markov assumption in simplifying the equations and there is substantial similarity to an HMM, this model is not strictly an HMM. Both distributions violate the conditional independence assumptions in a traditional HMM, as they are conditioned on the previous word, w_{t-1}.

In our system, the language model probability $P(w_t|s_t,w_{t-1})$ is obtained via an LM using class-based smoothing, as will be described further in Section 3. In BBN's work, this emission probability is obtained via a simple back-off bigram language model. In the original BBN model, each word is deterministically assigned one of 14 non-overlapping features, such as *two-digit-number*, *contains-digit-and-period*, *capitalized-word*, and *all-capital-letters*. When a bigram is infrequently observed, the back-off distribution depends on the assigned feature.

2.2 Model topology

The HMM-like topology of our model has several important characteristics. First, we wish to model both sentence boundaries and phrase boundaries, believing that there are important contextual effects associated with both. These boundaries are modeled explicitly in the topology, which includes sentence-initial and sentence-final states (necessitating the insertion of a pseudo-observation at each end of a word sequence). Also, each phrase type has two states associated with it: the first models the first word of the phrase, the second models all successive words. Phrase boundaries can thus be reliably determined, even when two phrases of the same type occur consecutively.

Figure 1 shows a simplified view of the topology. Note the pairs of states for each phrase type—in this case only those for *location* and *other* are shown in detail. In general, however, the second of each pair can only be reached from the first (and itself, via a self-loop), while the first can be reached from any other state except *end*.

The topology provides an implicit model of phrase length, as the self-loop transition probabilities to a state represent a form of geometric distribution of phrase lengths. For a certain phrase type, if p is the transition probability between the first and second state of the phrase model, and q is the self-loop

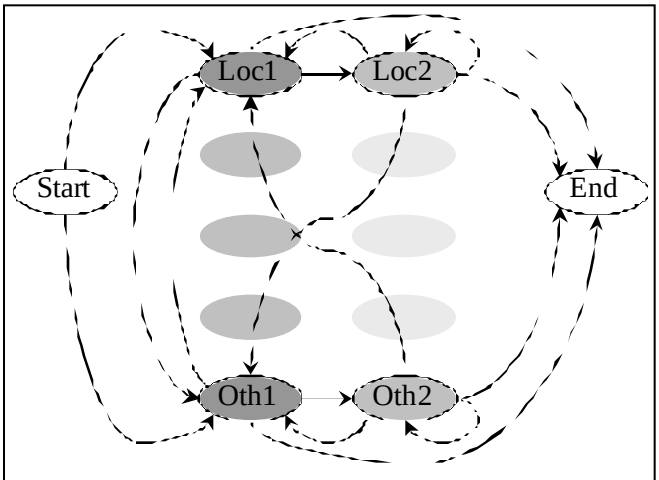

Figure 1: Simplified model topology

probability for the second state, the probability of a phrase of length l is as follows:

$$P(l) = 1 - p \qquad \text{for } l = 1$$
$$P(l) = pq^{(l-2)}(1-q) \qquad \text{for } l > 1$$

Modeling phrase length in this way is useful for syntactic phenomena such as proper names, in which one- and two-word phrases are very common.

2.3 Parameter estimation

One advantage of the model topology described in Section 2.2 is the fact that, given training data hand-labeled with phrase type and extent, the state corresponding to each word is completely observable. The first word in each phrase is emitted by the first state, while all remaining words are emitted by the second state. Consequently, the maximum likelihood values of the transition parameters can be easily calculated from corpus counts in the training data without the need for the Expectation-Maximization algorithm. However, because the state transitions are conditioned on the previous word, the model is susceptible to sparse data issues. Throughout our statistical model, we use linear interpolation to compensate for sparse data, smoothing the values with the lower order statistics. In the case of the state transitions, the interpolated formula becomes:

$$P(s_t \mid s_{t-1}, w_{t-1}) = \lambda P_{ML}(s_t \mid s_{t-1}, w_{t-1}) + (1-\lambda) P_{ML}(s_t \mid s_{t-1}).$$

P_{ML} is a maximum likelihood (relative frequency) estimate taken directly from training data, and λ is a word-dependent interpolation constant, type C from (Witten and Bell 1991). The value of λ is determined by the following formula:

$$\lambda = \frac{n(s_{t-1}, w_{t-1})}{n(s_{t-1}, w_{t-1}) + r(s_{t-1}, w_{t-1})}$$

where n() is the number of times a context occurs, and r() is the number of unique outcomes of that context.

3. STATE DEPENDENT LANGUAGE MODELS

In order to produce a robust model applicable to spoken language data, we wish to limit the dependence on the actual words in the text and on features derived from their orthographic realization. Unlike data from text sources, such as newspapers, the words in transcriptions that are output by a speech recognizer may not be the "correct" words. For that reason, it is helpful to use a smoother HMM observation distribution (i.e., the state-dependent language model) than would be estimated from accurately transcribed training material. Similarly, while orthography features—such as punctuation, capitalization, and the presence of non-alphabetic characters—provide useful information for distinguishing tokens in textual data, they are normally absent in speech data. For example, *$30.25* in text becomes "thirty dollars and twenty five cents" in speech transcriptions.

We address the need for more robust features by using a class-based smoothing technique (Iyer & Ostendorf, 1997), which has previously been used to develop speech recognition language models that successfully combine information from many sources. An advantage of this method is that it allows us to incorporate information from place and name word lists, as well as simple part-of-speech labels, and account for the fact that some words can be used in multiple classes. When available, the standard "clean text" features such as capitalization and punctuation can also be included in this way. In the following sections we will describe our language modeling approach in detail.

3.1. Class-based smoothing

In our class-based language model, rather than the simple bigram with back-off based on the words, the bigram probability is obtained by smoothing over the possible linguistic classes for the words in the bigram. This is shown in the following formula, where c_k ranges over the possible linguistic classes of word w_t:

$$P(w_t \mid w_{t-1}, s_t) = \sum_k P(w_t \mid c_k, w_{t-1}, s_t) P(c_k \mid w_{t-1}, s_t)$$

As in the case of the state transition probabilities discussed in Section 2.3, sparse data issues are addressed using linear interpolation with lower order statistics:

$$P(w_t \mid c_k, w_{t-1}, s_t) = \lambda P_{ML}(w_t \mid c_k, w_{t-1}, s_t) + (1-\lambda) P_{ML}(w_t \mid c_k, s_t)$$

Where the interpolation constant λ is defined analogously to that in Section 2.3. Training the language model consists of first assigning a part-of-speech tag to each word in the training data, then calculating the bigram statistics for the formula above.

In the first step of training, the words in the training data are labeled with part-of-speech information using the MITRE part-of-speech tagger, an implementation of the transformation-based learning approach introduced in (Brill 1992). The tagger assigns to each word one of 40 tags from the Penn Treebank tagset. The MITRE tagger has a reported accuracy of 93–95% on single-case text with no punctuation, which is similar to the speech transcriptions we are processing.

After POS tagging is completed, the training data is separated into its component "languages." For example, the PERSON language is estimated by creating a new file containing all the words (and corresponding POS tags) from PERSON phrases in the training data, with each phrase treated as a separate sentence or utterance. For each of these languages, a language model is trained according to the description above.

3.2. Use of word lists

A major consideration in developing information extraction systems in general is the treatment of unknown words. In most current speech recognition systems, the size and content of the system lexicon is predetermined, and the recognizer will output the word in its lexicon which most nearly matches the input audio stream. For the purposes of information extraction from the outputs of such systems, we only need to focus on the words contained in the system lexicon. In contrast, the problem of unknown words receives much attention in text data, where the vocabulary is unlimited. However, as speech systems evolve to being able to intelligently process words that are not explicitly in the lexicon, information extraction systems need to be more closely integrated with the recognizer. Rather than simply processing the words output from a "black box", we would like to be able to improve the treatment of unknown words and thereby improve the ability to extract information from the original audio signal.

For the purposes of identifying names in speech data, unknown words are a significant concern. While the overall out-of-vocabulary (OOV) rate is typically very low (<1%) for most large-vocabulary (48k–64k) recognition systems, the OOV rate is significantly higher for words in name phrases, frequently ranging from 5% to greater than 20%. In addition, the OOV rate can vary greatly depending on the type of name phrase, such that an unknown word is not equally likely to be found in all phrase types. Since a large part of our work focuses on training separate language models for the types of name phrases, being able to classify unknown words according to the type of name phrase is very important.

We accomplish this by classifying unknown training words into several tokens that are correlated with the types of name phrases. Our current system contains four such word tokens:

1. unknown person
2. unknown location
3. unknown person OR location
4. other unknown

The classification of the unknown words is determined by the presence of the words in word lists, which are independent of the main LM lexicon. We currently use two word lists, both obtained from public domain sources: a list of first and last names from the U.S. Census (~90,000 words) and a list of location names from the TIPSTER Gazetteer (~120,000 word). In this manner, the probability mass from the original unknown word token is distributed to tokens which better correlate with the types of phrases we are identifying.

In our initial experiments using word lists in this manner we obtained a 10% relative improvement in overall system performance. Our analysis of the output indicates that the wordlists are indeed helping the model identify and classify relevant unknown words. However, the use of such extensive lists has the disadvantage that it overgenerates NEs for ambiguous words such as Macarena (a popular dance, but also a city in South America).

4. EXPERIMENTS AND RESULTS

The system we describe above was formally evaluated under the auspices of the DARPA-sponsored Hub-4 (Broadcast News Transcription and Understanding) workshop in February 1999. The next section will describe this evaluation and our results. In addition to formal evaluation as part of the Hub-4 workshop, we ran several systematic experiments to determine the contributions of various system components to our NE performance. Sections 4.2–4.3 will describe these additional experimental results.

4.1. Hub-4 evaluation

In the official Hub-4 information extraction evaluation, participating sites were provided with a set of manually-annotated training data as well as a development test set. Sites used this data, either via machine learning or via manually-written rules, to develop systems which could automatically annotate data. The sites then ran their systems on several common test sets, and the outputs were scored against a manually-annotated "key" and compared. The test sets consisted of different transcriptions of the same news broadcasts: a manually-transcribed reference as well as three transcriptions from different speech recognizers. All participating sites produced named entity annotation for these four transcriptions, so the results can be directly compared. Each site additionally produced annotation for a fifth transcription, but since the fifth transcription was different for each site, the results cannot be directly compared.

The training, development and evaluation datasets we used were prepared for the Hub-4 IE evaluation by MITRE and SAIC. In addition, BBN prepared 100 hours of training data and made it available to the community for use in this evaluation. The combined training data sets from MITRE/SAIC and BBN

consisted of about one million words and about 50,000 named entities. The evaluation set consisted of about 32,000 words, 1800 named entities. Both the training and evaluation data consisted of a combination of American broadcast sources, both television and radio news, from a range of dates between 1996 and 1998.

We participated in all parts of the Hub-4 evaluation using the system described above. The lexicon consisted of 63k words representing a combination of the lexicons from two large-vocabulary speech recognition systems as well as a list of words from the MITRE part-of-speech tagger. As described in Section 3.2, the additional person and location name lists used in unknown word classification were obtained from public domain sources.

Table 1 shows our system results for each of the four common evaluation data sets. In addition to the common sets, we collaborated with SRI to produce annotation on their recognizer output, which had a WER of 21.1%.

For each of the ASR data transcriptions, our system results were as good or better than the other top systems evaluated. In addition, our model produced near-human results (88% accuracy) on the reference transcription (WER 0%).

WER (%)	F-Measure
28.3	71.1
21.1	78.1
14.5	82.2
13.5	81.6
0	88.2

Table 1: System performance for a range of word error rates.

4.2. Effect of training data size

Determining the effect of the amount of training data on task performance is important. While there are large amounts of training data available for the English NE task, similar amounts are not available for other information extraction tasks, including noun and verb phrase parsing and non-English NE. For such tasks, algorithms that deal well with data sparsity are desirable.

BBN has published results of experiments in which they trained their NE system with different amounts of training data, ranging from 100k words to over one million words. They found that overall performance in NE recognition increased in a log-linear fashion; that is, for each doubling in the amount of training data, the NE performance improved by a few points. We performed similar experiments with our system. We divided the Hub-4 training data into four subsets and created separate training sets from the subsets in all possible combinations. Evaluating the resulting systems on a separate test set produced log-linear results similar to those reported by BBN.

4.3. Contribution of class-based smoothing

Our objective in using class-based smoothing in the bigram language model was to produce more robust models than the simple bigram model, which is dependent exclusively on word identity. In order to determine the contribution of the class-based smoothing in the language model, we repeated the training set size experiments with language models trained without class information. For each of the training subsets, we retrained the language model, collapsing all POS tags to a single tag, and thus effectively removing the class smoothing. Testing these models on the ASR data with the highest word error rate (28.3%), we found that removing the POS information from the language models resulted in a consistent degradation (1–2% absolute) in named entity performance for all training sets.

While the class-based smoothing produced consistent improvement in errorful speech data, this improvement was not observed when the same models were used in annotating the reference transcripts with WER of 0%. However, as mentioned in the previous section, the training and evaluation data used consisted of more than one million words of broadcasts from a wide range of dates (1996–1998). The data also consisted of a mixture of broadcast domains: world news summaries such as CNN's *The World*, topical news shows such as ABC's *Nightline* and CSPAN's *Public Policy*, and radio news shows such as NPR's *All Things Considered* and *Marketplace*. The class-based smoothing has been the most effective in combining sparse data from multiple domain, as shown in (Iyer and Ostendorf, 1997). To demonstrate this effect using our phrase models for information extraction, we completed two further experiments on the reference transcript data. In both experiments we defined training and test sets that would have less overlap in content than the larger Hub-4 training and test data.

In the first experiment, we defined a new training set that consisted of the CNN world news broadcast programs in the Hub-4 training set; this set comprised 63 broadcasts with 365k words. We defined an independent test set from the Hub-4 data consisting of eight NPR broadcasts. The training data thus consisted entirely of broad coverage television news data while the test data consisted of topical radio news data. Comparing the performance of language models trained with and without class-based smoothing on this data indicated that

the class-based smoothing improved overall performance by 2.3% relative.

In the second experiment, we defined a training set consisting of the files from the Hub-4 data that represented broadcasts from early 1996, 40 broadcasts with 200k words. We defined an independent test set consisting of the 8 broadcasts from the middle of 1998 that represented the largest time difference between broadcasts in the data. Comparing the performance of language models trained with and without class-based smoothing on this data indicated that the class-based smoothing improved overall performance by 1.6% relative.

5. CONCLUSIONS

In this work we have introduced a robust framework for the labeling of linguistic structure in spoken language data, based on class-based smoothing. We have shown that the model yields consistently high performance on the task of recognizing named entities in transcriptions of spoken broadcast news data.

Our experiments indicate that for errorful speech data, class-based smoothing in a bigram language model produces a performance increase over standard bigram language models. This increase is consistent over a range of word error rates. In addition, for information extraction tasks for which smaller amounts of training data are available, or for which only out-of-domain training data sets are available, the class-based smoothing can also produce higher performance on reference transcriptions of speech data.

Our current system was developed with speech data in mind; consequently, we do not use capitalization or punctuation, even when they are present in the training data or reference transcripts. However, the modeling framework is extensible; it allows for the integration of any additional features, including features unique to text data, such as punctuation and capitalization. In addition, the extensible framework allows for the inclusion of additional features, such as word confidence scores and acoustic information. Similarly, while our initial language model implementation consists of class-based smoothing over part-of-speech categories, the classes we use are not limited to part-of-speech categories. They could also be automatically generated classes (via clustering), and the word lists can also be used to assign more specific class labels.

ACKNOWLEDGEMENTS

We would like to thank Rukmini Iyer for her help in understanding and modifying her language modeling code. We also thank BBN for preparing and releasing the additional NE training data and for providing insight into their Identifinder system. Finally, we thank SRI for making the output of their speech system available to us for the Hub-4 evaluation and providing other data for additional experiments.

REFERENCES

1. D. Bikel, S. Miller, R. Schwartz, R. Weischedel (1997), "NYMBLE: A High-Performance Learning Name Finder." *Proc. Applied Natural Language Processing*, 1997.

2. D. Bikel, R. Schwartz, R. Weischedel (1999), "An Algorithm that Learns What's in a Name." *Machine Learning*, in press.

3. E. Brill (1992), "A Simple Rule-based Part of Speech Tagger." *Proc. Third Conference on Applied Natural Language Processing*, Trento, Italy.

4. R. Iyer and M. Ostendorf (1997), "Transforming Out-of-Domain Estimates to Improve In-Domain Language Models." *Proc. Eurospeech*, 1997.

5. H. Witten and T.C. Bell (1991), "The Zero Frequency Estimation of Probabilities of Novel Events in Adaptive Text Compression." *IEEE Transactions on Information Theory*, vol. 7, no. 4, pp. 1085–1094, 1991.